REMARKS

Claims 1-4, 7-18 and 23-26 are currently pending and under examination. Claims 1, 11 15 and 23 are amended herein. Support for the amended claims can be found throughout the specification and claims as originally filed, for example, at page 7, lines 9-12, 16-17; page 7, line 30-page 8, line 27; page 10, lines 5-18; page 11, lines 6-14; page 26, lines 1-24; page 30, line 23 – page 31, line 10; page 31, line 23-page 34, line 23; and Fig 7. No new matter is added and entry of the present amendment is respectfully requested.

Interview Summary

The Applicant and Applicant's representative thank Examiner Stork for the courtesies extended in the interview held on February 14, 2006. Applicant believes the discussions were useful in identifying possible claim amendments that would better define the invention so as to facilitate allowance of the present application. In particular, Applicant pointed out several shortcomings of the *Rickards* and *Dean* references in the interview, for example, that the processes of these references are server based, limited to a static, rather than evolving, DTD, and do not teach pushing of edit and change events to each author in their systems as required in the present claims. The present amendments make explicit these requirements.

Claim Amendments

Claim 1 is amended to clarify certain aspects of the claimed invention and to advance prosecution. First, the claim amendments clarify that one purpose of a DTD is to define elements, their content models and each element's connectors (for example – | & ,) and occurrence indicators (for example – * + ?) in their respective content models. These content model modifiers produce an infinite number of DTD *variants* with no modification to the set of elements. In a static DTD, the set of elements does not change for each variant. In contrast, in the present invention, the ability to change the set of DTD elements in addition to being able to invoke the occurrence indicators to produce different DTD variants exists. Although not seeking to be limited to any particular theory of action, this evolving DTD emulates how natural language works. *See, e.g.*, specification at § 2.5 (Natural Language Imitation). The behavioral aspect of natural language imitation is emulated by code development which in the invention of

the present claims is tied to each element, so the change sequence is, for example: 1) declare the element, 2) declare its content model, and 3) define and deploy the code associated with that element. At any (run)time the element and or its code can be changed by its author and redeployed, deleted or perhaps further differentiated. This further differentiation could mean, for example, subdividing the semantic meaning of the element into sub-elements which further differentiates the meaning. Each of the sub-elements would subsequently acquire their own content models and code behavior. Thus, the DTD is continuously evolving.

Second, the claim amendments clarify that the present invention's presentation layer is strictly tied to its underlying grammar structure. *See, e.g.*, specification page 7, line 30-page 8 (regarding content navigation), line 4; page 30, line 23-page 31, line 10 (discussing the five styling cues of Figure 7 – indentation, numerical decimals, bold font, shading and read-only privileges – and their relation to underlying grammar structure); and Figure 7. A portion of an element's code development may involve programming these cues using common GUI programming techniques. Proper programming of these cues induces the effect in the author of a seamless connection to the underlying grammar structure without sacrificing single author look and feel. This induced effect serves to unambiguously navigate an author or viewer about the target work for synchronized viewing comprehension and cooperative editing purposes. In addition, context authors can edit the DTD directly. *See, e.g.*, specification page 7, lines 9-12. Thus, GUI cues and DTD structure are tightly coupled yet fully distributed, which is a key aspect that permits authors and subscribers to contemporaneously navigate and be navigated by the common continuously evolving DTD and presentation layer.

As indicated previously, change events occur when a node author completes a change to either his content or his context. For example, when a change is complete, all other participants receive and incorporate these changes as quickly as the intervening network allows in order to maintain an up-to-date replicate of the work product. There is no network pulling of change event data or polling for changes by the receiver. As a consequence, and in contrast to a pull-based network, regardless of how many participants a particular work product has, the network traffic among them is zero if no changes are being made to the distribute work product.

Withdrawn Rejections

The Applicant respectfully notes the withdrawal of the following rejections:

- the rejection of claims 1-2, 8-9, and 13-14 under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 6,061,697 (*Nakao*), in view of U.S. Patent No. 5,805,897 (*Glowny*);
- the rejection of claims 3 and 4 under 35 U.S.C. § 103 as obvious over *Nakao*, in view of *Glowny*, and further in view of U.S. Patent No. 5,918,010 (*Appleman*);
- the rejection of claims 5 and 6 under 35 U.S.C. § 103 as obvious over *Nakao*, in view of *Glowny* and *Appleman*, and further in view of U.S. Patent No. 5,764,731 (*Yablon*);
- the rejection of claim 7 under 35 U.S.C. § 103 as obvious over *Nakao*, in view of *Glowny*, and further in view of U.S. Patent No. 6,802,022 (*Olson*);
- the rejection of claims 10 and 12 under 35 U.S.C. § 103 as obvious over *Nakao*, in view of *Glowny*, and further in view of U.S. Patent No. 6,519,603 (*Bays*); and
- the rejection of claims 11 and 15-26 under 35 U.S.C. § 103 as purportedly obvious over *Nakao*, in view of *Glowny*, and *Bays*, and further in view of U.S. Patent No. 5,297,279 (*Bannon*).

Claim Objections

The Examiner has objected to claims 1-4, 7-18 and 23-26 for the recitation of the phrase "operate in a peer-to-peer environment without the need for a central server" in claim 1. Office action, page 2 ¶ 4. In particular, the Examiner objects to the recitation of the term "need" in this phrase. Claim 1 is amended herein in accordance with the Examiner's requirement.

Claim 11 is objected to for the recitation of the term "need." The Examiner has required correction of this claim. Accordingly, claim 11 is amended herein in accordance with the Examiner's requirement.

Claims 15 and 23 are objected to for the recitation of the phrase "wherein said context author can administrate." Office action, page 2 ¶ 6. The Examiner has required correction of

this claim. Accordingly, claims 15 and 23 are amended herein in accordance with the Examiner's requirement.

Rejection of Claims 1-4, 7-16 and 23-26 Under 35 U.S.C. § 103

Claims 1-4, 7-16 and 23-26 stand rejected under 35 U.S.C. § 103(a) as purportedly unpatentable over U.S. Publication No. 2002/0107994 (*Rickards*), in view of U.S. Publication No. 2002/0152244 (*Dean*). The Examiner has stated that *Rickards* discloses each aspect of the claims, with the exception of the DTD limitation. *Dean* is cited for teaching the use of a DTD. The motivation for combining *Rickards* with *Dean* advanced by the Examiner is that this combination "would have allowed a user to define conformance rules for document fragments." Applicant respectfully traverses this rejection.

As described below, both *Rickards* and *Dean* fail to teach certain aspects of the system of the present claims. For at least this reason the combination of these references fails to render obvious the system of the present claims.

I. *Rickards*

A. Server-Dependent and Contemporaneous Viewing

Rickards notes that its "Collaboration Engine 15 can be used as a stand alone program on a single computer . . . but can also be run on a computer as a server which can provide services to other computers that are connected to it via some connection such as . . . a local area network, etc." *See Rickards* ¶ 0087. This collaboration engine, which comprises an essential component of *Rickards* methods, is also referred to in *Rickards* as a "collaboration engine server" (CES). *See, e.g., id.* at Figures 1-2; ¶¶ 0019-0020, 0088 and 0108. *Rickards*, therefore, is clearly server-based.[1] Similarly, the "Collarboration Explorer" (CE) of *Rickards* (*see, e.g., Rickards* Figures 6 & 7) describes the exemplary work product ownership pattern. This process itself is a separate process from the work product. Technically, the CES exists in one computer process and the

[1] *See also* Rickards Fig. 22, which points out the need for a Notification Server. These two servers are necessary for Rickards because the main thrust of the invention is to provide a key-stroke capturing superstructure whereby existing applications can control parcelled out segment owners and authors.

work product segment exists in another (a computer process is usually equivalent to an application). Being separate processes means one process controls the other and since there are many segments, *Rickards'* CES functions as the server of the many client segments, hence *Rickards* is fundamentally client-server architecturally. Moreover, the CE "separate process," which is the implementation for viewing segment ownership in the work product, not being an integral part of the contemporaneously viewed work product, can therefore only offer secondary indications concerning a particular user's permitted work. Consequently, a *Rickards* user must notify other users (*see, e.g., Rickards* page 3, ¶ 0036) if editing has begun on a particular segment to avoid the pitfalls of multiple authors working in parallel on the same segment, each without the knowledge of the other author's edits. "Notification" in this context is not the same as incorporating a change directly into the work product and permitting contemporaneous viewing of the change by all authors. This shortcoming of *Rickards*, with regard to the present claims, is not overcome by mentioning the phrase "peer-to-peer" (*see id.* at page 4, ¶ 0052) with the term "client-server" (*see id.* at page 4, ¶ 0053), which merely indicates that the CES can co-exist on the same network configuration as any or all of the segments. In contrast, the fragment executables of the present claims are neither client nor server. *See, e.g.*, the present specification, Figures 3 & 5.

In sum, allowing more than one person to work on the same work product segment is implicit admission by *Rickards* that only collaborative, not contemporaneous, edit capability is provided. In contrast, the system of the present claims provides for cooperative and contemporaneous editing and obviates the need for after-the-fact contribution merging or reconciliation. Therefore, *Rickards* cannot operate without server processes, and does not allow multiple authors to navigate by graphical user interface cues and DTD context and edit a distributed document and DTD, and does not permit contemporaneous viewing of edits and DTD context changes.

B. Natural Language Imitation and Contemporaneous Editing

The system of the present claims operates with a semantic grammar (*see, e.g.*, specification § 4.1 "semantic context free grammar"), whereas *Rickards'* segmentation of work product does not. Designating work product segments in *Rickards* provides no language naming

component (*i.e.*, the action which produces semantic grammar elements) which is a primary aspect of the system of the present claims (*see, e.g.*, specification § 2.5 "Natural Language Imitation"). The underlying semantic grammar of the system of the present claims is strictly related to the presentation surface structure through graphical user interface techniques, emulating how natural language works (it may be noteworthy that strict deep-to-surface structural relationship is an accepted linguistic principle). It is this grammar-to-presentation construct that allows the underlying semantic grammar to be manipulated and to manipulate more than one user cooperatively and contemporaneously in the system of the present claims. *Rickards* offers no individual code development associated with its segments, as the presently claimed system provides. By definition, *Rickards* only controls the keystrokes that arrive at a single application implementation which means that all segment owners are fundamentally executing the same code with different ownership rights. Therefore, *Rickards* does not allow multiple authors to navigate by graphical user interface cues and DTD context and edit a distributed document and DTD. In contrast, in the present system, each semantic grammar node executable is to be developed and deployed upon creation. *See* specification at § 5.6.2 "Context Node Executable Generation"). This permits contemporaneous changing of DTD context and document editing by individual users of the presently claimed system.

II. ***Dean***

The examiner has cited the *Dean* reference for purportedly teaching an implemented DTD, which Rickards was admitted as failing to teach. *Dean*, however, utilizes a static, rather than evolving, DTD. For example, the first sentence of the Abstract indicates that a static DTD is utilized – "A method on an information processing unit performing steps for creating a user interface (UI) to assemble a document that conforms to a particular document type definition." In *Dean*, DTDs are retrieved and parsed (see, e.g., Figure 13A) before subsequent UI processing can occur. This means that a *Dean* DTD-to-UI iteration has to be conventionally versioned, as is required in all static DTDs. In the presently claimed system there is no one-time parsing of a DTD. Instead, in the evolving DTD of the present system, distributed micro-compilations occur each time a change event is completed, which can be a change to a DTD element or on the content of an element. These DTD context changes are pushed among the fragment executables, which permits each of the multiple authors and subscribers to view edits and DTD changes made

by others of the multiple authors contemporaneously. Versioning, by definition, is non-cooperative since all potential DTD grammar contributions are suppressed between versions and, additionally, must be funneled through a single source of DTD change in client-server fashion. *Dean* too, therefore, is server-dependent (see above discussion regarding the drawbacks and differences of server-dependent systems). *See also* Figures 1, 7-9 and 12 (and associated discussion); and ¶¶ 42-43, 88, 90, etc.

III. The Combination of *Rickards* and *Dean*

Based on the foregoing it should be clear that certain deficiencies exist in both the *Rickards* and *Dean* references such that, even when combined, this combination cannot render obvious the presently claimed system. For example, these references, either alone or in combination, fail to teach at least (1) a system that can operate without a central server; (2) an evolving DTD implemented by pushing edit and DTD context changes among fragment executables; (3) contemporaneous editing of a distributed document and DTD context changes; (4) contemporaneous viewing of edits and DTD changes; and/or (5) user navigation by graphical user interface cues and DTD context. Accordingly, the combination of *Rickards* with *Dean* fails to render the present claims obvious. Withdrawal of this rejection is respectfully requested.

Application No. 09/944,131 Docket No. 22339-40020

CONCLUSION

If, for any reason, the Examiner determines that the pending claims are not in condition for allowance, Applicant requests that the Examiner call the undersigned attorney at 202-736-8143 in an effort to resolve any matter still outstanding *before* the issuance of another action.

Favorable reconsideration is respectfully requested.

Applicant believes that no additional fee is due. However, if any fee under 37 C.F.R. §§ 1.16 or 1.17 should be due to render this response timely or otherwise to maintain status of the application as pending, the Commissioner is requested to charge such fee to our Deposit Account no. 18-1260 referencing docket number 22339-40020.

Respectfully submitted,

David L. Devernoe
Registration No. 50,128
Attorney for Applicant

SIDLEY AUSTIN LLP
1501 K Street, N.W.
Washington, D.C. 20005
Phone: 202-736-8143
Fax: 202-736-8711

Date: 2/17/2006

DC1 775650v.2